An Analysis of Backward Simulation Data-Refinement for Partial Relation Semantics

Moshe Deutsch and Martin C. Henson
Department of Computer Science, University of Essex, UK.
{mdeuts, hensm}@essex.ac.uk

Abstract

This paper investigates data-refinement by backward simulation for specifications whose semantics is given by partial relations. The standard model-theoretic approach is based on totalisation and lifting. The paper examines this model, exploring and isolating the precise roles played by lifting and totalisation in the standard account by introducing a simpler, normative theory of backward simulation data-refinement (SB-refinement) which captures refinement directly in the language and in terms of the natural properties of preconditions and postconditions. This theory is used in conjunction with four other model-theoretic approaches to determine the extent to which the standard approach is canonical, and the extent to which it is arbitrary.

1. Introduction

Refinement underlies the transformational software process model, in which design decisions are incorporated into an initial abstract specification deriving, in stages, more concrete versions. In data-refinement, the objective is to transform a data type into a form closer to an implementation: the underlying data space is refined along with the operation. This process is sometimes called data design [20]. There are two refinement techniques which enable us to verify such transformations: forward simulation and backward simulation [18, 5]. These are known to be sound and jointly complete [10, 19] and can also be formulated as theories of refinement in their own right.

In this paper, we investigate data-refinement by backward simulation for specifications whose semantics is given by partial relations.

In order to formulate a theory of data-refinement for this underlying semantics, the standard approach is as described in [18] (Chapter 16, et seq.). This is a model-theoretic approach, in which the operations (partial relations) are both completed (made total) and extended (by means of an additional semantic value). This modelling also involves extending the simulation relations, though not completing them. As is often the case with a model-based approach, the technical machinery raises a number of questions: why must the specifications and simulations be extended? Why extended in the manner presented? Why must the specifications be completed in the manner presented? In other words, one is interested in ensuring that there is a very clear motivation for what might otherwise appear to be rather arbitrary technical choices.

In this paper, we consider six data-refinement theories, confining attention to refinement induced by backward simulation. These constitute generalisations of various operation refinement theories explored in [7] and [8] of which two are related to previous work [18, 5]. No systematic investigation or results concerning the relationships between them, have been presented or published before. We will prove that four of these theories are equivalent and demonstrate that five of them are acceptable as refinement characterisations. In exploring these issues we integrate a mathematical with an informal analysis, using a variety of examples and counterexamples. Our results shed some light on the standard model-theoretic approach, in particular precisely explaining the extent to which it is canonical and the extent to which it is arbitrary.

We begin by introducing the notion of data simulation that underlies the forward and backward simulation refinement techniques (section 2), including the lifted simulations used in refinement based on relational completion operators (discussed in, for example, [18, 5] and investigated in detail in [7, 8]). These involve an additional distinguished element, called bottom and written ⊥. We then define three alternative characterisations of data-refinement (section 3) based on two distinct relational completion models discussed in [7] (see also section A.3). We show that all three

1 That work was, however, undertaken explicitly in Z, whereas this paper is concerned with partial relations in general. Z is the most well-known example of partial relation semantics, in which schemas denote sets of bindings which establish, in general, a partial relation between before and after states. Our previous work is based on ZC, the logic for Z reported in [13].
are equivalent to a purely proof theoretic characterisation of backward simulation refinement (section 4). This fourth theory, SB-refinement, captures backward simulation data-refinement directly in terms of the language, the relationship between the data types involved, and the concept of precondition. It is a more abstract, less constructive notion, not involving the introduction of either an auxiliary semantics, nor the introduction of an auxiliary element. We regard it as the normative theory for exploring the validity of refinement approaches that are based on backward simulation.

Our approach sheds light on the role of lifting in data-refinement based on relational completion models by investigating a backward simulation refinement characterisation based on the non-lifted totalisation discussed in [7] (see also section A.4). We show that this is an unacceptable refinement theory (section 5) and explain the reasons why the non-lifted totalisation is an adequate model for operation refinement [7] but not data-refinement, emphasising the significance of \( \perp \) in ensuring validity of model-theoretic approaches (section 6).

We employ a novel technique of rendering all the theories of refinement as sets of introduction and elimination rules. This leads to a uniform and simple method for proving the various equivalence results. As such, it contrasts with the more semantic based techniques employed in [3]. We summarise some notational conventions and definitions in appendix A.

2. Data Simulations

The methods of data-refinement in state-based systems are well established. The conditions under which a transformation is a correct refinement step can be summarised by two simulation based refinement techniques: forward simulation and backward simulation [4]. In this section we revise these and introduce some essential material underlying our investigation.

A data simulation [18, 20] is a relation between an abstract data space and a concrete counterpart. Data simulations\(^2\) are derived from two refinement techniques which enable us to verify data-refinement, as shown by the two semi-commuting diagrams in Fig. 1. Both forward and backward simulation\(^3\) refinement techniques are known to be sound but neither of them is singly complete. However, they are known to be jointly complete [19].

\(^2\) The notion of simulation is overloaded in the literature. Various authors use it to denote a certain refinement technique, whereas others use it to denote the retrieval relation used in a certain refinement technique. In this paper we use the word “simulation” to specifically denote a retrieval relation.

\(^3\) Forward and backward simulations are also respectively known as downward and upward simulations [4, 5, 11] due to their directions in the commuting diagrams in Fig. 1.

![Fig.1. Forward simulation and backward simulation refinement techniques.](image)

Forward Simulation

Backward Simulation

\[
\begin{align*}
\text{Forward Simulation:} & \\
S & \subseteq U \\
\text{Backward Simulation:} & \\
S & \subseteq U
\end{align*}
\]

We will use the meta-variables\(^4\) \( U_0 \) and \( U_1 \) to range over our specifications (binary relations). In this paper \( U_0 \) will always be the concrete operation and \( U_1 \) the abstract operation. We adopt the approach taken in [3]: our concrete relation is always drawn from \( \mathcal{P}(T_0 \times T_0) \) and the abstract relation from \( \mathcal{P}(T_1 \times T_1) \). A backward simulation (concrete to abstract) belongs to \( \mathcal{P}(T_0 \times T_1) \).

We will need to incorporate the \( \perp \) element in a simulation used with lifted-totalised operations (see appendix A and [7, 8]). Naturally, Woodcock’s chaotic totalisation [18] is unacceptable here, as this might enforce a link between abstract and concrete states that are not supposed to be linked. The conventional approach [18, 5] is to (non-strictly) lift\(^5\) \( \perp \) in the input set of the simulation, thus retaining its partiality. This leads to the following definition:

**Definition 1 (Non-Strictly Lifted Backward Simulation).**

\[
\hat{S} =_{df} \{ (z_0, z_1) \in T_0 \times T_1 \mid z_0 \neq \perp \Rightarrow (z_0, z_1) \in S \}
\]

Then the following introduction and elimination rules are derivable:

**Proposition 1.**

\[
\frac{
\langle t_0, t_1 \rangle \in T_0 \times T_1, \quad t_0 \neq \perp \quad \langle t_0, t_1 \rangle \in S
}{
\langle t_0, t_1 \rangle \in \hat{S}_{(S)\text{\footnotesize \textsuperscript{\textcircled{\textcircled{o}}}}} (S^\text{\footnotesize \textsuperscript{\textcircled{o}}})
}\]

\[
\frac{
\langle t_0, t_1 \rangle \in \hat{S}_{(S)\text{\footnotesize \textsuperscript{\textcircled{\textcircled{o}}}}} (S^\text{\footnotesize \textsuperscript{\textcircled{o}}})
}{
\langle t_0, t_1 \rangle \in T_0 \times T_1, \quad \langle t_0, t_1 \rangle \in \hat{S}_{(S)\text{\footnotesize \textsuperscript{\textcircled{\textcircled{o}}}}} (S^\text{\footnotesize \textsuperscript{\textcircled{o}}})
}\]

\[^4\] We provide some notational conventions in appendix A.

\[^5\] Lifting signifies mapping \( \perp \) of the input set of the relation onto all states of the output set. In general, the notion of strictness discussed in this paper is with respect to \( \perp \); therefore, strict lifting denotes mapping \( \perp \) onto only its output counterpart.
Lemma 1. The following additional rules are derivable for non-strictly lifted simulations:

\[
\begin{align*}
S \subseteq S & \quad (i) \\
\langle \perp, \perp \rangle \in S & \quad (ii) \\
\langle \perp, t \rangle \in S & \quad (iii) \\
\frac{t \in T_{1_0}}{T_0 \models \perp} & \quad (iv)
\end{align*}
\]

\[\square\]

Lemmas 1(i – iv) demonstrate that definition 1 is consistent with the intentions described in [18] and [5]: the underlying partial relation is contained in the lifting; the \(\perp\) element is mapped onto every after state, and no other initial state is so. This raises an immediate question: why does the lifting of the simulation have to be non-strict with respect to \(\perp\)? This issue was not explored in [18, 5], where the non-strict lifting of the simulation is taken as self-evident. We will gradually provide an answer to this question in the sequel. For that, we will need the definition of a strictly lifted simulation:

Definition 2 (Strictly Lifted Backward Simulation).

\[\tilde{S} \equiv \{ (z_0, z_1) \in T_{0_0} \times T_{1_0} \mid (z_0 \neq \perp \Rightarrow \exists z_1 \exists z_1 : (z_0, z_1) \in S) \land (z_0 = \perp \Rightarrow z_1 = \perp) \}\]

Obvious introduction and elimination rules follow from this.

Lemma 2. The following additional rules are derivable for strictly lifted simulations:

\[
\begin{align*}
S \subseteq S & \quad (i) \\
\langle \perp, \perp \rangle \in S & \quad (ii) \\
\langle \perp, t \rangle \in S & \quad (iii) \\
\frac{\langle t_0, t_1 \rangle \in \tilde{S}}{t_0 = \perp} & \quad (iv) \\
\frac{\langle t_0, t_1 \rangle \in \tilde{S}}{t_1 \neq \perp} & \quad (v)
\end{align*}
\]

\[\square\]

Lemmas 2(iv – v) embody the strictness captured by definition 2: if the after state is \(\perp\) then the initial state must also be \(\perp\), and if it is not \(\perp\) then the initial state was not either.

3. Data-Refinement with Backward Simulation

In [7] and [8] we investigated operation refinement (that is the degenerate case of data refinement in which simulations are identity functions) for specifications whose semantics is given by partial relation semantics (though we used Z explicitly as an example in that earlier work). We compared three characterisations of operation refinement: S-refinement, a proof theoretic characterisation closely connected to refinement as introduced by Spivey [15]; W*-refinement, based on Woodcock’s relational completion operator [18]; and W₀-refinement based on a strict relational completion operator (see appendix A). We proved that all these refinement theories are equivalent. The investigation also illuminated the crucial role of \(\perp\) in total correctness operation refinement.

In this section, we provide four distinct notions of data-refinement, generalising the above operation refinement characterisations, based on backward simulation. We will then go on to compare them thus providing a complementary investigation to that in [7] and [8] in the more general setting of data refinement and partial relation semantics.

3.1. SB-Refinement

In this section, we introduce a purely proof theoretic characterisation of backward simulation refinement, which is closely connected to sufficient refinement conditions introduced by Woodcock [18, p.270] (indicated as “B-corr”) and by Derrick and Boiten [5, p.93]. These conditions correspond to the premises of our introduction rule for SB-refinement.

This generalisation of S-refinement [7, 8] is based on two properties expected in a refinement: that postconditions do not weaken (we do not permit an increase in non-determinism in a refinement) and that preconditions do not strengthen (we do not permit requirements in the domain of definition to disappear in a refinement). In this case these two properties must hold in the presence of a simulation.

The notion can be captured by forcing the refinement relation to hold exactly when these conditions apply. SB-refinement is written \(U_0 \triangleright_{sb} U_1\) \((U_0 \text{ SB-refines } U_1)\) with respect to the simulation \(S\)^6 and is given by the definition that leads directly to the following rules:

Proposition 2. Let \(x, x_0, x_1, z, z_0\) be fresh variables.

\[
\begin{align*}
\langle x, z \rangle & \in S \quad \text{Pre } U_1 z \triangleright \text{Pre } U_0 x \\
\langle z_0, z \rangle & \in S \quad \text{Pre } U_1 z, \langle x_0, x_1 \rangle \in S, \langle z_0, x_0 \rangle \in U_0 \triangleright \langle z_0, t \rangle \in S \\
\langle x_0, z \rangle & \in S \quad \text{Pre } U_1 z, \langle x_0, x_1 \rangle \in S, \langle z_0, x_0 \rangle \in U_0 \triangleright \langle t, x_1 \rangle \in U_1 \\
U_0 \triangleright_{sb} U_1 & \quad (\triangleright_{sb}^+) \\
U_0 \triangleright_{sb} U_1 & \quad (\langle t, z \rangle \in S \rightleftharpoons \text{Pre } U_1 z \triangleright \text{Pre } U_0 t) \\
U_0 \triangleright_{sb} U_1 & \quad \langle t_0, t_2 \rangle \in S \quad \langle t_0, t_1 \rangle \in U_0 \\
U_0 \triangleright_{sb} U_1 & \quad \langle t_0, y \rangle \in S \quad (\langle t_1, y \rangle \in U_1 \triangleright P) \\
\end{align*}
\]

The usual sideconditions apply to the eigenvariable \(y\). \(\square\)

^6 We will omit the superscript \(S\) from now on, in this and other notions of refinement that depend upon a simulation.
This theory does not depend on, and makes no reference to, the \( \bot \) value. We take SB-refinement as normative: this is our prescription for data-refinement, and another theory is acceptable providing it is at least sound with respect to it.

3.2. Relational Completion Based Refinement

We now introduce three backward simulation refinement theories. These are based on the two distinct notions of the schema lifted-totalisation set out in appendix A. Each of them captures, schematically, the backward simulation commuting diagram in Fig. I and is based on relational composition.

**WB\_Refinement.** This notion of refinement is also discussed in [18, p.247] and [4]. It is written \( U_0 \models_{wb_u} U_1 \) and is defined as follows:

**Definition 3.** \( U_0 \models_{wb_u} U_1 =df \) \[ \langle z_0, z_1 \rangle \in U_0 \models S \Rightarrow \langle z_0, z_1 \rangle \in S \models U_1 \]

The following introduction and elimination rules are immediately derivable for WB\_refinement:

**Proposition 3.** Let \( z_0, z_1 \) be fresh.

\[
\begin{align*}
\langle z_0, z_1 \rangle \in U_0 \models S & \Rightarrow \langle z_0, z_1 \rangle \in S \models U_1 \\
U_0 \models_{wb_u} U_1 & \Rightarrow \langle t_0, t_1 \rangle \in U_0 \models S \Rightarrow \langle t_0, t_1 \rangle \in S \models U_1
\end{align*}
\]

\( \Box \)

**WB\_Refinement.** The natural generalisation of W\_Refinement [7] (at least in the light of the standard literature) is to use strict-lifted totalised operations, yet a non-strict lifted simulation. We name this WB\_Refinement, written \( U_0 \models_{wb} U_1 \) and defined as follows:

**Definition 4.** \( U_0 \models_{wb} U_1 =df \) \[ \langle z_0, z_1 \rangle \in U_0 \models S \Rightarrow \langle z_0, z_1 \rangle \in S \models_U U_1 \]

Obvious introduction and elimination rules follow from this.

**WB\_Refinement.** Our third characterisation of refinement is motivated by the enquiry raised in section 2. Establishing a refinement theory, in which both the operations and the simulation are strictly lifted, provides a point of reference which will aid us in investigating two important matters: firstly, whether the strict and non-strict relational completion operators are still interchangeable underlying generalisations of data-refinement; secondly, whether the non-strict lifting of the simulation is an essential property. We name this theory WB\_Refinement, written \( U_0 \models_{wb_u} U_1 \); it is defined as follows:

**Definition 5.** \( U_0 \models_{wb_u} U_1 =df \) \[ \langle t, z \rangle \in S \models \neg S \subseteq S \models \neg U_1 \]

Obvious introduction and elimination rules follow from this definition.

4. Four Equivalent Theories

In this section, we demonstrate that all four theories of refinement are equivalent and we will clearly see the critical role that the \( \bot \) value plays in the three model-theoretic approaches.

We shall be showing that all judgements of refinement in one theory are contained among the refinements sanctioned by another. Such results can always be established proof-theoretically because we have expressed even our model-theoretic approaches as theories. Specifically, we will show that the refinement relation of a theory \( T_0 \) satisfies the elimination rule (or rules) for refinement of another theory \( T_1 \). Since the elimination rules and introduction rules of a theory enjoy the usual symmetry properties, this is sufficient to show that all \( T_0 \)-refinements are also \( T_1 \)-refinements. Equivalence can then be shown by interchanging \( T_0 \) and \( T_1 \).

4.1. WB\_Refinement and SB-Refinement are Equivalent

We begin by showing that WB\_refinement implies SB-refinement, by proving that WB\_refinement satisfies both SB-refinement elimination rules. Firstly the rule for preconditions.

**Proposition 4.** The following rule is derivable:

\[
U_0 \models_{wb_u} U_1 \Rightarrow \langle t, z \rangle \in S \models \text{Pre } U_1 z \Rightarrow \text{Pre } U_0 t
\]

**Proof**

\[
\begin{align*}
\delta_0 & : \\
\vdots & : \\
\langle t, \bot \rangle & \in S \models U_1 \text{ false} \quad (2) \\
\text{false} & \Rightarrow \text{Pre } U_0 t \quad (1)
\end{align*}
\]

Where \( \delta_0 \) stands for the following branch:

\[
\begin{align*}
\neg \text{Pre } U_0 t \quad (1) & \Rightarrow \langle t \in T_0 \rangle \quad (T) \\
\langle t \in T_0 \rangle & \Rightarrow \langle t \in T_0 \rangle \quad (L, 4(iii)) \\
\langle t, \bot \rangle & \in U_0 \quad \text{false} \Rightarrow \text{Pre } U_0 t \quad (1)
\end{align*}
\]

\[
\begin{align*}
\langle t, \bot \rangle & \in U_0 \models S \Rightarrow \langle t, \bot \rangle \in U_0 \models S \\
\langle t, \bot \rangle & \in S \models U_0 \models U_1
\end{align*}
\]
and $\delta_1$ is:

\[
\begin{align*}
\langle t, y \rangle &\in S \quad (T) \\
\frac{i \in T_0}{\langle t, y \rangle \in S} &\quad (i \in T_0 
\downarrow \n\n\n
\frac{\langle y, \bot \rangle \in U_1}{\langle y, \bot \rangle \in U_1 (T)} \quad (2)
\end{align*}
\]

$\square$

First, the proof contains steps labelled (T): these indicate points at which we appeal to fact that the sets underlying operations and simulations do not contain the $\bot$ value. Naturally this is a necessary assumption without which the technical matter of lifting sets would fail. We discuss this a little more in appendix A. We will continue to label any proof steps appealing to this property in the same way in the proofs that follow.

Second, notice the explicit use of $\bot$ in the proof. This is reminiscent of our previous investigation of operation refinement in Z, in which the explicit use of $\bot$ is critical for proving that $W_{\ast}$-refinement satisfies the precondition elimination rule for $S$-refinement (see, for example, proposition 4.11 in [7]). Much the same observation can be made here, only that the use of lemmas 4(iii) and 1(ii) in the proof suggests that both the lifted-totalisation of the operations and the lifting of the simulation are essential for showing that $W_{\ast}$-refinement guarantees that preconditions do not strengthen in the presence of the simulation.

Turning now to the second elimination rule in SB-refinement.

**Proposition 5.** The following rule is derivable:

\[
\begin{align*}
\langle t_0, z \rangle &\in S \quad (S) \\
\langle t_1, t_2 \rangle &\in S \quad \langle t_0, t_1 \rangle &\in U_0 \\
\langle t_0, y \rangle &\in S, \langle y, t_2 \rangle &\in U_1 + P \\
\vdash &\quad \text{Pre } U_1 z \\
U_0 &\models_{ab} U_1 &\quad \text{Proof}
\end{align*}
\]

\[
\begin{align*}
\langle t_0, t_1 \rangle &\in U_0 \\
\langle t_1, t_2 \rangle &\in U_0 \\
\vdash &\quad \text{Pre } U_1 z \\
\langle t_0, t_1 \rangle &\in U_0 \\
\langle t_1, t_2 \rangle &\in S \\
\frac{\langle t_0, t_1 \rangle &\in U_0 \quad \text{(L, 4(iii))}}{\langle t_0, t_2 \rangle &\in U_0 + S} &\quad \text{(L, I(iii))}
\end{align*}
\]

\[
\begin{align*}
\langle t_0, t_2 \rangle &\in U_1 + P \\
\langle t_0, t_2 \rangle &\in S + U_1 \\
\langle t_0, t_2 \rangle &\in S + U_1 (T)
\end{align*}
\]

$\delta_0$ stands for the following branch:

\[
\begin{align*}
\delta_1 &\vdash \\
\langle t_0, t_1 \rangle &\in U_0 \\
\langle t_1, t_2 \rangle &\in S \\
\delta_0 &\vdash \\
\langle t_0, t_2 \rangle &\in S + U_1 \\
\delta_0 &\vdash \\
\langle t_0, t_2 \rangle &\in S + U_1 \\
\delta_0 &\vdash
\end{align*}
\]

Where $\delta_1$ is:

\[
\begin{align*}
\langle t_0, y \rangle &\in S \quad (I) \\
\langle t_0, t_1 \rangle &\in U_0 \quad \text{(T)} \\
\langle t_0, y \rangle &\in S \quad (I) \\
\langle t_0, t_1 \rangle &\in U_0 \quad \text{(T)} \\
\langle y, t_2 \rangle &\in U_1 \\
\langle y, t_2 \rangle &\in U_1 \\
\Rightarrow &\quad \text{Pre } U_1 y
\end{align*}
\]

$\square$

**Theorem 1.** $U_0 \supseteq_{ab} U_1 \Rightarrow U_0 \supseteq_{ab} U_1$

**Proof.** This follows immediately, by $(\supseteq_{ab}^S)$, from propositions 4 and $S^\ast$. $\square$

We now show that SB-refinement satisfies the $W_{\ast}$-elimination rule.

**Proposition 6.**

\[
\begin{align*}
U_0 &\models_{ab} U_1 \\
\langle t_0, t_1 \rangle &\in U_0 + S \\
\langle t_0, t_1 \rangle &\in S + U_1 \\
\text{Proof}
\end{align*}
\]

\[
\begin{align*}
\text{Proof } \phi \text{ be: } &\quad \forall z \bullet \langle t_0, z \rangle \in S \Rightarrow \text{Pre } U_1 z \quad \forall z \bullet \langle t_0, z \rangle \in S \land \neg \text{Pre } U_1 z
\end{align*}
\]

\[
\begin{align*}
\Rightarrow &\quad \text{Pre } U_1 z \\
\Rightarrow &\quad \text{Pre } U_1 z \\
\Rightarrow &\quad \text{Pre } U_1 z
\end{align*}
\]

Where $\delta_0$ stands for the following branch:

\[
\begin{align*}
\langle t_0, t_1 \rangle &\in U_0 \quad \text{(I)} \\
\langle t_0, y \rangle &\in U_0 \quad \text{(I)} \\
\langle t_0, y \rangle &\in U_0 \quad \text{(I)} \\
\Rightarrow &\quad \text{Pre } U_1 t_0 \\
\text{Proof } \phi \text{ be: } &\quad \forall z \bullet \langle t_0, z \rangle \in S \Rightarrow \text{Pre } U_1 z \quad \forall z \bullet \langle t_0, z \rangle \in S \land \neg \text{Pre } U_1 z
\end{align*}
\]

Where $\delta_0$ stands for the following branch:

\[
\begin{align*}
\langle t_0, t_1 \rangle &\in U_0 \quad \text{(I)} \\
\langle t_0, t_1 \rangle &\in U_0 \quad \text{(I)} \\
\langle t_0, t_1 \rangle &\in U_0 \quad \text{(I)} \\
\Rightarrow &\quad \text{Pre } U_1 t_0 \\
\Rightarrow &\quad \text{Pre } U_1 t_0 \\
\Rightarrow &\quad \text{Pre } U_1 t_0
\end{align*}
\]

Where $\beta_0$ is:

\[
\begin{align*}
\langle t_0, t_1 \rangle &\in \text{Pre } U_1 t_0 \\
\langle t_0, t_1 \rangle &\in \text{Pre } U_1 t_0 \\
\langle t_0, t_1 \rangle &\in \text{Pre } U_1 t_0
\end{align*}
\]

and $\beta_1, \beta_2$ are respectively:

\[
\begin{align*}
\beta_0 &\vdash \\
\beta_1 &\vdash \\
\beta_2 &\vdash
\end{align*}
\]

$\text{The proofs of such theorems are always automatic by the structural symmetry between introduction and elimination rules. We shall not give them in future.}$
\[ \langle t_0, w_0 \rangle \in S \] 
\[ (L, 1(i)) \quad \langle w_0, t_1 \rangle \in U_1 \] 
\[ (3) \]
\[ \langle t_0, w_0 \rangle \in S \] 
\[ (L, 1(i)) \quad \langle w_0, t_1 \rangle \in U_1 \] 
\[ (3) \]
\[ \delta_1 \text{ stands for the following branch:} \]
\[ \exists z \cdot \langle t_0, z \rangle \in S \land \neg \mathit{Pre} U_1 z \]
\[ \langle t_0, t_1 \rangle \in S \upharpoonright U_1 \] 
\[ (2) \]
\[ \langle t_0, t_1 \rangle \in S \upharpoonright U_1 \] 
\[ (4) \]
\[ \text{Where } \alpha_0 \text{ is:} \]
\[ \langle t_0, w_1 \rangle \in S \land \neg \mathit{Pre} U_1 w_1 \] 
\[ \langle t_0, t_1 \rangle \in S \] 
\[ (L, 1(i)) \]
\[ \langle w_1, t_1 \rangle \in U_1 \] 
\[ (4) \]
\[ \langle t_0, t_1 \rangle \in S \upharpoonright U_1 \] 
\[ \text{and } \alpha_1 \text{ is:} \]
\[ \neg \mathit{Pre} U_1 w_1 \] 
\[ \langle t_0, w_1 \rangle \in S \] 
\[ \langle y, t_1 \rangle \in T_0 \times T_1 \] 
\[ (y, t_1) \in S \] 
\[ \langle w_1, t_1 \rangle \in U_1 \] 
\[ (4) \]
\[ \langle t_0, t_1 \rangle \in S \upharpoonright U_1 \] 
\[ \text{and } \alpha_2 \text{ is:} \]
\[ \langle t_0, w_1 \rangle \in S \land \neg \mathit{Pre} U_1 w_1 \] 
\[ \langle w_1, t_1 \rangle \in U_1 \] 
\[ (4) \]
\[ \langle t_0, w_1 \rangle \in S \] 
\[ (5) \]
\[ \langle w_1, t_1 \rangle \in U_1 \] 
\[ (L, 4(iv)) \]
\[ \text{Notices that this proof depends on use of the law of excluded middle (see, for example, [17, p.47] and [7, p.105]). We suspect that this result is strictly classical, and there appear to be many other examples of this in refinement theory.} \]

**Theorem 2.** \( U_0 \models_{a} U_1 \Rightarrow U_0 \models_{a} U_1 \)

**Theorem 3.** \( U_0 \models_{a} U_1 \Rightarrow U_0 \models_{a} U_1 \)

A very similar situation arises when we consider \( W_{b} \)-refinement. \( S \)-refinement constitutes our common ground and again we need to make use of the substitutions and amendments to the proofs of propositions 4, 5 and 6 as we did in theorem 3 except that \( \exists_{w_{b}} \) replaces \( \exists_{w_{b}} \). \( S \) replaces \( S \) and we apply lemmas 2(i) and 2(iii) in place of lemmas 1(i) and 1(ii) (respectively). Moreover, applications of \( (\gamma_{0}) \) and \( (\gamma_{f}) \) replace \( (\gamma_{0}) \) and \( (\gamma_{f}) \) (respectively). From this we immediately get implication in both directions:

**Theorem 4.** \( U_0 \models_{a} U_1 \Rightarrow U_0 \models_{a} U_1 \)

Despite their superficial dissimilarity, all four theories are equivalent. In establishing this we reinforce the results from [7] and [8] showing clearly the significance of \( \perp \) (proposition 4). Additionally we have shown that strict lifting of both the operations and the simulation is sufficient for introducing a model based refinement theory that preserves the natural properties of \( S \)-refinement.

The fact that, given the appropriate substitutions, the proofs in this section are identical to the ones in section 4.1 suggests that the minimal mathematical properties of the models, which are essential for establishing theorems 1 and 2 are the ones of \( U \) and \( \tilde{S} \). To be more specific, the use of lemma 4(iii) (proposition 4) indicates that everything outside the preconditions of the underlying operation, including \( \perp \), should be mapped onto \( \perp \) of the output set; and the use of the proof step labelled (**) in conjunction with lemma 4(iv) (proposition 6) indicates that everything outside the preconditions that is not \( \perp \) should be mapped onto everything in the output set. These observations are precisely the properties of strictly-lifted totalised relations within a non-strict framework. A similar observation can be made for the simulation: the only lemma concerning the lifting of the simulation used in the proofs is 1(iii) (proposition 4); there is no evidence for requiring lemma 1(iii), which expresses the non-strict lifting.

### 5. The Non-Lifted Totalisation underlying Data-Refinement

In [7] we presented \( \mathit{Pre} \) as a distinct semantics for the notion of the precondition of an operation. This notion underlies \( W_{e} \)-refinement, a model-theoretic operation refinement theory based on non-lifted totalisation\(^8\) (denoted \( \hat{U} \))

\(^8\) The definitions of \( \mathit{Pre} \) and the non-lifted totalisation can be found in section A.4. Moreover, in this section, we will refer to the standard definition of preconditions as \( \mathit{Pre} \) in order to distinguish it from the new notion.
of the underlying operations. We demonstrated that \( W_r \)-refinement is equivalent to \( S_1 \)-refinement, a normative characterisation of refinement which is identical to \( S \)-refinement [7, 8] with all occurrences of \( \text{Pre}_0 \) substituted by \( \text{Pre}_1 \). This allows us to obtain an acceptable model-theoretic characterisation of operation refinement, without having to use \( \perp \) values.

In this section we will show that this is not the case under the generalisation to data-refinement, highlighting the inevitability of using \( \perp \) values in both lifting (of the simulation) and relational completion models (of the operations) underlying backward simulation refinement.

We begin by introducing \( W_b \)-refinement as a generalisation of \( W_r \)-refinement [7] with backward simulation. Since the totalisation of the operations is not lifted, we do not lift the simulation either; nor do we totalise it for reasons discussed in section 2. Therefore, \( W_b \)-refinement is defined as follows:

**Definition 6.** \( U_0 \models_{wb} U_1 \iff U_0 \triangleright S \subseteq S \triangleright U_1 \)

Obvious introduction and elimination rules for \( W_b \)-refinement follow from this definition.

The definition above raises an immediate question: is \( W_b \)-refinement an acceptable theory of refinement? if not what are the reasons for that? In [7] we concluded that the (chaotic) lifted-totalisation and the non-lifted totalisation coincide in a “\( \perp \)-less” framework, under the interpretation of \( \text{Pre}_1 \). However, that followed noting that \( W_r \)-refinement is an acceptable refinement theory: it is sound (as well as complete) with respect to the normative theory \( S_1 \)-refinement. Clearly we need to take the same approach here: \( S_1 \)-refinement (that is SB-refinement with all instances of \( \text{Pre}_0 \) substituted by \( \text{Pre}_1 \) would be our normative characterisation, guaranteeing the two properties expected in a backward simulation refinement (section 3.1), under the interpretation of \( \text{Pre}_1 \). Notice that we can prove that SB-refinement satisfies both \( S \)-refinement elimination rules. This is a straightforward consequence of lemma 6 (section A.4). From this we immediately get the following theorem:

**Theorem 5.** \( U_0 \models_{wb} U_1 \Rightarrow U_0 \models_{wb} U_1 \square \)

We start with completeness: \( W_b \)-refinement is complete with respect to \( S_1 \)-refinement since \( S_1 \)-refinement satisfies the elimination rule for \( W_b \)-refinement.

**Proposition 7.** The following rule is derivable:

\[
\begin{align*}
U_0 \models_{wb} U_1 \quad \langle \langle t_0, t_1 \rangle \rangle \in U_0 \triangleright S \\
\langle \langle t_0, t_1 \rangle \rangle \in S \triangleright U_1
\end{align*}
\]

The structure of the proof is very similar to proposition 6. \( \square \)

Then the following theorem is immediate:

**Theorem 6.** \( U_0 \models_{wb} U_1 \Rightarrow U_0 \models_{wb} U_1 \square \)

An example of this is given in Fig. 2, where each of the diagrams constitutes an extension of the backward simulation commuting diagram in Fig. 1, showing the (completed) operations and the (lifted in case of \( W_b \)-refinement) simulation. We can use \( W_b \)-refinement to represent an \( S_1 \)-refinement, in such examples, because we have theorem 5 and we know that \( S \)-refinement and \( W_b \)-refinement are equivalent (theorems 1, 2). We can observe that both dia-

![Fig.2. An example: WB\(_\ast\)-refinement is complete with respect to SB\(_\ast\)-refinement](image)

grams illustrate a classic case of weakening the precondition in the presence of backward simulation: \( t_1 \) is outside the precondition of the abstract operation, yet its concrete counterpart \( t_0 \) is in the precondition of the concrete operation, and they are both linked by the simulation. This property is, naturally, sanctioned by \( S_1 \)-refinement and, due to theorem 6, also by \( W_b \)-refinement. However, does \( W_b \)-refinement guarantee that preconditions do not strengthen? In order to answer this, we need to investigate whether \( W_b \)-refinement is sound with respect to \( S_1 \)-refinement.

Employing the same proof strategy involving elimination rules, we start by proving that \( W_b \)-refinement satisfies the \( S_1 \)-elimination rule for postconditions.

**Proposition 8.** The following rule is derivable:

\[
\begin{align*}
\langle t_0, z \rangle \in S \vdash \text{Pre}_1 U_1 z \\
\langle t_0, t_1 \rangle \in S \quad \langle t_0, t_1 \rangle \in U_0 \\
U_0 \models_{wb} U_1 \quad \langle \langle t_0, t_2 \rangle \rangle \in S \quad \langle t_0, t_2 \rangle \in U_1 \vdash P \\
P
\end{align*}
\]

**Proof**

\[
\begin{align*}
\langle t_0, t_1 \rangle \in U_0 & \quad \text{(L. 7(i))} \\
\langle t_0, t_1 \rangle \in U_0 \quad \langle t_1, t_2 \rangle \in S \\
U_0 \models_{wb} U_1 \quad \langle \langle t_0, t_2 \rangle \rangle \in U_0 \triangleright S \\
\langle t_0, t_2 \rangle \in S \triangleright U_1 & \quad \delta \\
\vdots & \quad \vdots \\
\langle t_0, t_2 \rangle \in S \triangleright U_1 & \quad P
\end{align*}
\]
Where $\delta$ stands for the following branch:

$$
\begin{align*}
(t_0, y) \in S & \quad (I) \\
\langle y, t_2 \rangle \in \bar{U}_1 & \\
\langle t_0, y \rangle \in S \land \langle y, t_2 \rangle \in \bar{U}_1 & \\

\end{align*}
$$

From this we can deduce that $W_{B_1}$-refinement guarantees that postconditions do not weaken. Nevertheless, it cannot guarantee that preconditions do not strengthen because it fails to satisfy the $SB_1$-elimination rule for preconditions. If we attempt to prove proposition 4 with $\exists_{wb}$ replacing $\exists_{wb}$ and $Pre_{i}$ in place of $Pre_0$, we immediately learn that, unlike the proof of proposition 4, we cannot derive a contradiction from the assumption that $t$ is not in the precondition of $U_0$. This is precisely because the non-lifted totalisation does not involve $\perp$ values which, ultimately, lead to the contradiction required in the proof of proposition 4. We exhibit a counterexample in Fig. 3, which manifests this observation. Both diagrams capture the data we have in the proof.

\[ WB_{B_1}\text{-refinement } \Rightarrow \text{ WB}_{B_1}\text{-refinement} \]

![Fig. 3. A counterexample: WB\textsubscript{$B_1$}-refinement is not sound with respect to SB\textsubscript{$B_1$}-refinement](image)

In conclusion, the fact that $W_{B_1}$-refinement is not sound with respect to $S\text{B}_1$-refinement makes it an unacceptable refinement. We would like to highlight the significance of using a normative refinement characterisation as a common ground in such investigations: it enables us to pinpoint the source of the problem in terms of the two basic properties concerning preconditions and postconditions and to construct representative counterexamples, which illustrate the problem (e.g. Fig. 3).

### 6. Discussion

The non-lifted totalisation underlying refinement introduces a variety of problems. Woodcock [18, p.237-238] motivates an explanation to some extent. We discussed that thoroughly in [7] (section 4.4), where we, ultimately, raised a question: why is there a distinction between implicit (Chaos) and explicit (True) permission to behave in a lifted totalised framework and not in a non-lifted totalised one? In this section we will gradually answer this question, and secure the observations we made in section 5.

A useful way to examine the essence of a relational completion model is by scrutinising it under extreme specifications (see, for example, [9], chapter 3). This enables us to observe and explain phenomena that might not emerge otherwise. In this spirit we define two such specifications which respectively denote explicit and implicit “permission to behave”:

**Definition 7.** (i) $True = \forall \exists \{ z \in T \times T \mid true \}$
(ii) Chaos $= \forall \exists \{ z \in T \times T \mid false \}$

By applying the (chaotic) lifted-totalisation and the non-lifted totalisation to these specifications we immediately get the following:

**Lemma 3.** (i) $True \neq Chaos$ (ii) $True \neq Chaos$ □

Lemma 3(i) represents a counterexample, in which augmentation of undefinedness is possible in a refinement based on non-lifted totalisation, under the standard interpretation of preconditions (section A.2). This is remedied by $W_{\ast}$-refinement [7] because the Woodcock-completion [18] imposes a distinction between implicit and explicit permission to behave collapses, leads to a Woodcock-like operation refinement theory, $W_{\ast}$-refinement [7], in which the relations need not be lifted. This theory is simply defined as a subset relation of the (non-lifted) totalised relations, where the subset prevents augmentation of nondeterminism and the non-lifted totalisation, in conjunction with the subset, plays the same role as its lifted counterpart in preventing augmentation of undefinedness. Therefore, $W_{\ast}$-refinement is an acceptable refinement theory that guarantees these two elementary properties without utilising $\perp$ values. So why does the non-lifted totalisation have no future underlying model-theoretic refinement? More specifically, we are asking: why does it not work for data-refinement?

---

9 Here we take $T \times T$ to be the universe of relations.
The answer concerns the data simulations and the properties of composition. Consider the example in Fig. 4, where we present the specification True\(^1\) as a refinement of a certain specification \(U_1\), under both WB\(^-\)-refinement and WB\(^+\)-refinement. This is not a case of weakening the postcondition because the simulation links the first output state in \(U_1\) with all the output states in True\(^1\); thus, it is a sensible case of data-refinement. Yet in a non-lifted totalised operation, it is impossible to indicate whether an input state is mapped onto all output states as a result of not being mapped onto anything, or being mapped onto everything in the underlying operation. For this reason, the specifications True and Chaos are indistinguishable in this model and therefore the WB\(^-\)-refinement case in Fig. 4 also holds for Chaos, as we can see in Fig. 5. Naturally it is unacceptable that a chaotic specification refines some other specification that is not chaotic. This means that undefinedness has been augmented, as a result of strengthening preconditions. Indeed, as we have seen in section 5, WB\(^-\)-refinement sanctions this feature and is therefore unacceptable as a refinement theory. This case is prohibited by WB\(^+\)-refinement, precisely because the lifted-totalisation maps input states outside the precondition of the underlying operation onto \(\bot\), as well as everything else in the output set. Thus, WB\(^-\)-refinement fails in Fig. 5 since all the highlighted paths leading to \(\bot\) are not associated with paths in the other direction. Notice that the only way to establish these paths is through a link between \(t\) and \(\bot\) in \(U_1\): this does not exist, because \(t\) is in the precondition of \(U_1\).

In conclusion, \(\bot\) underlies the distinction between True and Chaos in the lifted totalised framework. This prevents imprudent cases of refinement such as the one in Fig. 5 by prohibiting a strengthening of preconditions in the presence of the simulation. For this reason we prefer to refer to \(\bot\) simply as the “distinguished” value, rather than “undefined” [18] or “non-termination” [9, 14, 2, 12], or even our own previous suggestion, the “abortive” value [7].

7. Conclusions and Future Work

In this paper, we introduced six distinct notions of data-refinement founded upon backward simulation. By reformulating these as theories, rather than as sufficient conditions, we established a mathematical framework, which underlies our analysis. We demonstrated in section 4 that what look like different models of specification and refinement are, in fact, intimately related. Having a non-model-theoretic benchmark (SB-refinement) allows us to scrutinise the role of the \(\bot\) value in the model-theoretic approaches and evaluate the essence of the relational lifted-totalisation found in the literature.

The SB-refinement theory is entirely proof-theoretic, characterising refinement directly in terms of the language and the behaviour of preconditions and two basic observations regarding the properties one expects in a refinement: preconditions do not strengthen and postconditions do not weaken in the presence of backward simulation. We advocate a different approach to [18] and [5] by taking SB-refinement to be the fundamental characterisation of refinement, rather than (what we have denoted as) WB\(^-\)-refinement. Such an approach has two major advantages: first, we establish a clear normative framework based on unquestionable properties. We have seen in section 5 that whenever a potential theory fails to be sound with respect to the normative theory, we can pinpoint the grounds for that failure, in terms of the two basic properties concerning preconditions and postconditions. This aids us in isolating the problem and in constructing representative counterexamples that illuminated relational completion, in general, and the non-lifted totalisation in particular. Secondly, as we reported in section 4, having a normative theory for investigating the relationships amongst various candidate theories

---

\(^{10}\) Here, as usual, True belongs to \(\wp(T_0 \times T_n)\) and \(U_1\) to \(\wp(T_1 \times T_n)\).

\(^{11}\) Recall that, although WB\(^-\)-refinement is unacceptable as a refinement theory, it still guarantees that postconditions do not weaken (prop. 8). This is certainly the case for WB\(^+\)-refinement, being as it is, equivalent to SB-refinement.
not only simplifies the process (for example, as we have seen: many similarities in the proofs), it also enables us to compare the details of the proofs. In this paper we have been led to the conclusion that the strict and non-strict relational completion models are interchangeable in the context of backward simulation refinement and so are the strict and non-strict lifted simulations.

In this paper we have not provided an analysis of forward simulation data-refinement. The results in this paper cannot be taken as self-evidently analogous in a counterpart investigation of forward simulation theories. Indeed, we show in [7] that forward simulation refinement is less permissive: the strict and non-strict relational completion models are still interchangeable in this framework, but the strict lifting of the simulation has a restrictive effect: WF∗-refinement (the forward simulation counterpart of WB∗-refinement) is sound with respect to SF-refinement (the normative theory in this case), but it is not complete because, under certain circumstances, it prevents strengthening of preconditions.

The non-lifted totalisation has the same impact in forwards refinement: WF∗-refinement (the forward simulation counterpart of WB∗-refinement) is an unacceptable refinement theory because it permits strengthening of preconditions and is, therefore, not sound with respect to the normative theory SF1-refinement.

There is much more to say about data-refinement, particularly generalising results we have detailed in [7] in the context of operation refinement, e.g. formulating forward and backward simulation refinement theories based on a sets of implementations model or data-refinement theories based on weakest preconditions, and then exploring their relationships with SB and SF-refinement (the bulk of this work is reported in [6]). There is an additional interesting dimension, in which we explore generalisations of firing conditions refinement [8,5,16] underlying forward and backward simulation techniques. We can investigate their relationships with a variety of data-refinement theories based on the abortive relational completion model as given in [8], [5] and [1].

8. Acknowledgements

Moshe Deutsch is supported by the British Council through an ORS award. This work has been influenced in its development by too many people to name explicitly. However, special thanks for particularly important discussions and comments go to Steve Reeves, Ray Turner, Eerke Boiten, John Derrick, Steve Dunne, Lindsay Groves, Ralph Miarka, Greg Reeve, Steve Schneider, David Streater, Helen Treharne, Mark Utting, Jim Woodcock and Rob Arthan.

References


A. Appendix

A.1. Notation and Basic Definitions

In this appendix, we will revise a little relational algebra, setting some notational conventions in the process.

First of all, we write an ordered pair drawn from $A \times B$ as usual: $(a, b)$ when $a \in A$ and $b \in B$.

We introduce relational composition via introduction and elimination rules, in keeping with our proof-theoretic approach. The composition of two relations $U_0 \subseteq U_1$ belongs to $\mathcal{P}(T_0 \times T_1)$ where, as expected, $U_0$ belongs to $\mathcal{P}(T_0 \times T_0)$ and $U_1$ belongs to $\mathcal{P}(T_2 \times T_1)$.

**Proposition 9.** The following rules are derivable (the usual side-conditions apply to the eigenvariable $y$):

\[
\frac{\langle t_0, t_2 \rangle \in U_0 \quad \langle t_2, t_1 \rangle \in U_1 \quad \langle t_0, t_1 \rangle \in U_0 \circledcirc U_1}{\langle t_0, t_1 \rangle \in U_0 \circledcirc U_1 \quad \langle t_0, y, t_1 \rangle \in U_1 + P (U_0 \circledcirc U_1)}
\]

\(\square\)

We must, of course, postulate that all underlying sets of the relations we consider are $\bot$-free: that is the value $\bot$, when we need to introduce it for the purposes of lifting, is a fresh constant. This ensures that the following definition is never degenerate.

**Definition 8.** $T_\bot \equiv T \cup \{\bot\}$

A.2. Preconditions

We can formalise the idea of the precondition of a specification as the domain of the relation it corresponds to. We, for brevity, let $U \in \mathcal{P}(T \times T)$.

**Definition 9.** $\text{Pre } U \ z \equiv \exists x \in T \bullet (z, x) \in U$

We now review the chaotic relational completion operators discussed in [7] and [8].

A.3. Lifted Totalisation

Firstly, we define the non-strict-lifted totalisation in line with the intentions described in [18], chapter 16. We will write $T$ for the set $T_0 \times T_1$.

**Definition 10.** $\mathcal{U} \equiv \{\langle z_0, z_1 \rangle \in T \mid \text{Pre } U \ z_0 \Rightarrow \langle z_0, z_1 \rangle \in U\}$

Then the following introduction and elimination rules are derivable:

**Proposition 10.**

\[
\frac{\langle t_0, t_1 \rangle \in T \quad \text{Pre } U \ t_0 \ + \ \langle t_0, t_1 \rangle \in U}{\langle t_0, t_1 \rangle \in \mathcal{U} \quad \langle t_0, t_1 \rangle \in \mathcal{U} \quad \langle t_0, t_1 \rangle \in \mathcal{U}}
\]

\(\square\)

**Lemma 4.** The following extra rules are derivable for lifted-totalised sets:

\[
\begin{align*}
\text{(i)} & \quad U \subseteq U \quad \langle \bot, \bot \rangle \in U \quad \langle t, \bot \rangle \in U \\
\text{(ii)} & \quad \neg \text{Pre } U \ t_0 \quad t_0 \in T_0 \quad t_1 \in T_1 \\
\text{(iii)} & \quad \langle t_0, t_1 \rangle \in U
\end{align*}
\]

\(\square\)

The strict-lifted totalisation is defined as follows:

**Definition 11.**

\[
\mathcal{S} \equiv \{\langle z_0, z_1 \rangle \in T \mid (\text{Pre } U \ z_0 \Rightarrow \langle z_0, z_1 \rangle \in U) \land (z_0 = \bot \Rightarrow z_1 = \bot)\}
\]

We obtain obvious introduction and elimination rules, which in this case we will not state explicitly. In addition, we have fairly standard properties:

**Lemma 5.**

\[
\begin{align*}
\text{(i)} & \quad U \subseteq U \\
\text{(ii)} & \quad \langle \bot, \bot \rangle \in U \\
\text{(iii)} & \quad \langle t, \bot \rangle \in \mathcal{U} \\
\text{(iv)} & \quad \langle t_0, t_1 \rangle \in U \\
\text{(v)} & \quad \langle t_0, t_1 \rangle \in \mathcal{U}
\end{align*}
\]

Notice that in (v) $t_0$ ranges over the unextended set. \(\square\)

A.4. Non-Lifted Totalisation

In [7], we defined a non-lifted totalisation operator based on a revised notion of preconditions. According to this notion, an input state is in the precondition of the operation if and only if it is mapped (by the operation) onto some output state(s) but not onto all of them. We name this $\text{Pre }\circledcirc$, given by the following definition:

**Definition 12.**

\[
\text{Pre }\circledcirc \ U \ z \equiv \exists x_0, x_1 \in T \bullet (z, x_0) \notin U \land (z, x_1) \in U
\]

It is evident that the new notion of preconditions implies the standard one.

**Lemma 6.** $\text{Pre }\circledcirc U \ t \Rightarrow \text{Pre }_0 U \ t \ \square$

Then the non-lifted totalisation of a set is given by the following definition.

**Definition 13.** $\mathcal{U} \equiv \{\langle z_0, z_1 \rangle \in T \times T \mid \text{Pre }\circledcirc U \ z_0 \Rightarrow \langle z_0, z_1 \rangle \in U\}$

Obvious introduction and elimination rules follow from this definition. Notice that the values of each ordered pair in this completion range over the natural set $T$, thus $\bot$ values do not play any role here.

**Lemma 7.** The following extra rules are derivable for non-lifted totalised sets:

\[
\begin{align*}
\text{(i)} & \quad U \subseteq U \\
\text{(ii)} & \quad \langle \bot, \bot \rangle \in U \\
\text{(iii)} & \quad \langle t_0, t_1 \rangle \in U
\end{align*}
\]

\(\square\)

\(^{13}\) In this section we refer to the standard definition of preconditions as $\text{Pre }_0$ in order to distinguish it from the new one.